ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-921

Preference Inference Based on Lexicographic Models

Nic Wilson[1]

Abstract. With personalisation becoming more prevalent, it can often be useful to be able to infer additional preferences from input user preferences. Preference inference techniques assume a set of possible user preference models, and derive inferences that hold in all models satisfying the inputs; the more restrictive one makes the set of possible user preference models, the more inferences one gets. Sometimes it can be useful to have an adventurous form of preference inference when the input information is relatively weak, for example, in a conversational recommender system context, to give some justification for showing some options before others. This paper considers an adventurous inference based on assuming that the user preferences are lexicographic, and also an inference based on an even more restrictive preference model. We show how preference inference can be efficiently computed for these cases, based on a relatively general language of preference inputs.

1 INTRODUCTION

User preferences are becoming increasingly important in intelligent systems, as personalisation becomes more prevalent. It is only rarely practical for a user to explicitly express all their preferences; it is thus important to be able to infer additional preference between outcomes from the user inputs. To do this one must make some kind of assumptions about the user model, specifically, about the form their preference relation takes. Approaches such as CP-nets and generalisations [3, 5, 4, 24, 1] make weak assumptions, that the user preference relation is a total order or a total pre-order. This is cautious, but has the disadvantage of leading to rather weak inferences (and also has high computational complexity [16]). In many cases it can be advantageous to make stronger assumptions about the user preference model, in order to be able to infer more preferences. For instance, in the context of conversational recommender systems, one is looking for reasons to prefer some options over others, in order to choose which options to show the user next.

More explicitly, the idea behind preference inference is as follows: we have a set of user preference inputs, and we assume a form of model of the user preference relation, leading to a set $\mathcal{F}$ of all such candidate preference relations. We then infer a preference $\alpha \geq \beta$, of one outcome over another, if $\alpha \succcurlyeq \beta$ holds for all preference relations $\succcurlyeq$ in $\mathcal{F}$ that satisfy the inputs. The inputs can be atomic preferences, i.e., preferences of one outcome (alternative) over another; or they can be preference statements which imply (often exponentially many) atomic preferences, such as used in CP-nets and more general preference languages [3, 17, 23, 24, 1]. Another possible input expresses the relative importance of different variables or sets of variables [5, 4, 25]. Preference inference can be considered, for certain situations, as an alternative to preference learning e.g., [15, 9, 13, 6, 2] where one is instead typically trying to choose a best model (within a particular family of models).

In Bridge and Ricci [7], for example, it is assumed that the preference relation on boolean variables is generated by a sum of unary functions (as in Multi-Attribute Utility Theory [10]), which we call here an *Additive Utility model*. User inputs then generate linear constraints on the unknown weights, and a linear programming solver is used to deduce further implied preferences, assuming this model.

Another choice of preference model is made in [23], where the preference relation is assumed to be a form of conditional lexicographic order; it is shown that a polynomial algorithm can then be used for preference input. This is applied in a conversational recommender system context in [21, 20].

When the preference inputs are especially sparse, an even more adventurous inference, arising from an even more restrictive model, may well be desirable. It is natural to consider models based on lexicographic orders [11, 12, 14], which correspond to both Additive Utility models and conditional lexicographic models.

The main focus of this paper is showing that preference inferences can be made efficiently, based on the family of lexicographic models, and with inputs being of a rather general form. The basic computational approach is similar to the preference inference for conditional lexicographic models in [23]. We also consider an even more restrictive family of preference models, which we call the singleton lexicographic, where a model is a total order on the domain of a single variable. This leads to additional inferences over lexicographic inference. We show that this is computationally relatively simple, and how it relates to lexicographic preference inference, and that it can be given a simple sound and complete proof theory

Section 2 considers lexicographic preference inference. Section 3 describes the singleton lex(icographic) inference. In Section 4 we show how lexicographic preference inference can be computed in low-order polynomial time, and Section 5 concludes.

Proofs are included in a longer version of the paper available online [26].

2 LEXICOGRAPHIC INFERENCE

We describe in this section the preference language on which we focus, and formally define lexicographic inference. We begin with some basic definitions.

Throughout the paper we consider a fixed set V of n variables. For each $X \in V$ let $D(X)$ be the set of possible values of X. For subset of variables $A \subseteq V$ let $\underline{A} = \prod_{X \in A} D(X)$ be the set of possible assignments to set of variables A. For $X \in V$, we abbreviate $\underline{\{X\}}$ to $\underline{X}$; it is in one-to-one correspondence with $D(X)$. An *outcome* is an element of $\underline{V}$, i.e., an assignment to all the variables. If $a \in \underline{A}$ is an assignment to A, and $b \in \underline{B}$, where $A \cap B = \emptyset$, then we may write

[1] Insight Centre for Data Analytics, School of Computer Science and IT, University College Cork, Ireland, email: nic.wilson@insight-centre.org. Also: School of EEE&CS, Queen's University Belfast, Belfast, UK.

ab as the assignment to $A \cup B$ which combines a and b. For partial tuples $a \in \underline{A}$ and $u \in \underline{U}$, we may say a *extends* u, if $A \supseteq U$ and $a(U) = u$, i.e., a projected to U gives u.

2.1 Preference Formulae and their projections to Y

In this paper we will consider preference statements φ of the form $p \geq q \parallel T$, where P, Q and T are subsets of V, with $(P \cup Q) \cap T = \emptyset$, and $p \in \underline{P}$ is an assignment to P, and $q \in \underline{Q}$ is an assignment to Q. Informally, the statement $p \geq q \parallel T$ represents that p is preferred to q if T is held constant, i.e., any outcome α extending p is preferred to any outcome β that extends q and agrees with α on variables T. Formally, the semantics of this statement is given by the relation φ^* which is defined to be the set of pairs (α, β) of outcomes such that α extends p, and β extends q, and α and β agree on T: $\alpha(T) = \beta(T)$.

Let $\mathcal{L}$ be the set of all preference statements φ of the form $p \geq q \parallel T$, as defined above. As shown in [23], this is a relatively expressive preference language. For outcomes α and β, a preference of α over β can be represented as $\alpha \geq \beta \parallel \emptyset$, which we abbreviate to just $\alpha \geq \beta$. Preferences between partial tuples can also be represented. *Ceteris Paribus* statements have $T = V - (P \cup Q)$, allowing, for example, the representation of feature vector rules [19]. The language can also express stronger statements of the form defined in [24], which generalises CP-nets and TCP-nets [3, 5]. (However, the representation of conditional preferences is not important when considering lexicographic inference: see Proposition 2 below).

For $\Gamma \subseteq \mathcal{L}$, we define Γ^* to be equal to $\bigcup_{\varphi \in \Gamma} \varphi^*$. This is the set of preferences between outcomes directly implied by Γ.

Projections to Y

The computational techniques below in Sections 3.1 and 4.3 make use of projections of preference statements to a single variable.

Let $\mathfrak{R} \subseteq \underline{V} \times \underline{V}$, let $Y \in V$ be a variable, and let $A \subseteq V - \{Y\}$ be a set of variables not containing Y. Define $\mathfrak{R}^{\downarrow Y}$, the projection of $\mathfrak{R}$ to Y, to be $\{(\alpha(Y), \beta(Y)) : (\alpha, \beta) \in \mathfrak{R}\}$. Also, define, $\mathfrak{R}^{\downarrow Y}_A$, the A-restricted projection to Y, to be the set of pairs $(\alpha(Y), \beta(Y))$ such that $(\alpha, \beta) \in \mathfrak{R}$ and $\alpha(A) = \beta(A)$. $\mathfrak{R}^{\downarrow Y}_A$ is the projection to Y of pairs that agree on A. Thus, $\mathfrak{R}^{\downarrow Y} = \mathfrak{R}^{\downarrow Y}_{\emptyset}$.

For comparative preference statement φ and set of comparative preference statements Γ we abbreviate $(\varphi^*)^{\downarrow Y}_A$ to $\varphi^{\downarrow Y}_A$ and abbreviate $(\Gamma^*)^{\downarrow Y}_A$ to $\Gamma^{\downarrow Y}_A$. We have $\Gamma^{\downarrow Y}_A = \bigcup_{\varphi \in \Gamma} \varphi^{\downarrow Y}_A$.

Together with the following result, this implies that $\Gamma^{\downarrow Y}_A$ can be computed efficiently.

Proposition 1 *Consider any element $\varphi = p \geq q \parallel T$ of $\mathcal{L}$. Let A be a set of variables and let Y be a variable not in A. If p and q are incompatible on A (i.e., $p(P \cap Q \cap A) \neq q(P \cap Q \cap A)$) then $\varphi^{\downarrow Y}_A$ is empty. Otherwise, $\varphi^{\downarrow Y}_A$ consists of all pairs $(y, y') \in \underline{Y} \times \underline{Y}$ such that (i) $y = y'$ if $Y \in T$; (ii) $y = p(Y)$ if $Y \in P$; and (iii) $y' = q(Y)$ if $Y \in Q$. (Thus if $p(P \cap Q \cap A) = q(P \cap Q \cap A)$ and none of conditions (i), (ii) and (iii) hold, then $\varphi^{\downarrow Y}_A = \underline{Y} \times \underline{Y}$.)*

2.2 Lexicographic models and inference

Define $\mathcal{G}^{\mathrm{lex}}$ to be the set of lexicographic models (over the set of variables V), where a lexicographic model, π (over V), is defined to be a sequence $(Y_1, \geq_{Y_1}), \ldots, (Y_k, \geq_{Y_k})$, where Y_i ($i = 1, \ldots, k$) are different variables in V, and each $\geq_{Y_i}$ is a total order on $\underline{Y_i}$. The associated relation $\succcurlyeq_\pi \subseteq \underline{V} \times \underline{V}$ is defined by, for outcomes α and β, $\alpha \succcurlyeq_\pi \beta$ if and only if either (i) for all $i = 1, \ldots, k$, $\alpha(Y_i) = \beta(Y_i)$; or (ii) there exists $i \in \{1, \ldots, k\}$ such that for all $j < i$, $\alpha(Y_j) = \beta(Y_j)$ and $\alpha(Y_i) >_{Y_i} \beta(Y_i)$ (i.e., $\alpha(Y_i) \geq_{Y_i} \beta(Y_i)$ and $\alpha(Y_i) \neq \beta(Y_i)$). Thus $\succcurlyeq_\pi$ is a total pre-order, which is a total order if $k = n = |V|$.

Lexicographic inference

For $\mathfrak{R} \subseteq \underline{V} \times \underline{V}$, and lexicographic model $\pi \in \mathcal{G}^{\mathrm{lex}}$, we define $\pi \models \mathfrak{R} \iff \succcurlyeq_\pi \supseteq \mathfrak{R}$, i.e., if $\alpha \succcurlyeq_\pi \beta$ for all $(\alpha, \beta) \in \mathfrak{R}$. For preference statement $\varphi \in \mathcal{L}$, we define $\pi \models \varphi \iff \pi \models \varphi^*$ i.e., for $(\alpha, \beta) \in \varphi^*$, $\alpha \succcurlyeq_\pi \beta$. For $\Gamma \subseteq \mathcal{L}$, define $\pi \models \Gamma \iff \pi \models \Gamma^*$, which is if and only if for all $\varphi \in \Gamma$, $\pi \models \varphi$. This leads to the definition of inference relation $\models^{\mathrm{lex}}$: for $\Gamma \subseteq \mathcal{L}$ and $\alpha, \beta \in \underline{V}$, $\Gamma \models^{\mathrm{lex}} \alpha \geq \beta \iff \alpha \succcurlyeq_\pi \beta$ holds for all $\pi \in \mathcal{G}^{\mathrm{lex}}$ such that $\pi \models \Gamma$.

We define $\succcurlyeq^{lex}_\Gamma$ to be the relation on outcomes thus induced by Γ, so that $\alpha \succcurlyeq^{lex}_\Gamma \beta \iff \Gamma \models^{\mathrm{lex}} \alpha \geq \beta$. The relation $\succcurlyeq^{lex}_\Gamma$ contains relation Γ^* and is a pre-order, since it is the intersection of a set of (total) pre-orders, i.e., $\{\succcurlyeq_\pi : \pi \models \Gamma\}$.

Note that, although the language $\mathcal{L}$ allows the expression of conditional preferences, the conditional part is irrelevant for $\models^{\mathrm{lex}}$ inference: in particular, we can write a statement $\varphi \in \mathcal{L}$ in a unique way in the form $ur \geq us \parallel T$, where $u \in \underline{U}$, $r \in \underline{R}$, $s \in \underline{S}$, and U, T and $R \cup S$ are (possibly empty) mutually disjoint subsets of V, and for all $X \in R \cap S$, $r(X) \neq s(X)$. Then, for any lexicographic model $\pi \in \mathcal{G}^{\mathrm{lex}}$, $\pi \models \varphi$ if and only if $\pi \models \bar{\varphi}$, where the associated unconditional preference statement $\bar{\varphi} \in \mathcal{L}$ is defined to be $r \geq s \parallel T \cup U$. For $\Gamma \subseteq \mathcal{L}$, define $\bar{\Gamma}$ to be $\{\bar{\varphi} : \varphi \in \Gamma\}$. This implies the following.

Proposition 2 *For any $\Gamma \subseteq \mathcal{L}$, and outcomes $\alpha, \beta \in \underline{V}$,* $\Gamma \models^{\mathrm{lex}} \alpha \geq \beta \iff \bar{\Gamma} \models^{\mathrm{lex}} \alpha \geq \beta$.

Relationship with cp-tree-based inference: In [23], a preference inference $\models_{\mathcal{Y}}$ is defined where the models are "$\mathcal{Y}$-cp-trees", which are a kind of generalised lexicographic order (similar to a search tree for solving a CSP), where both value and variable orderings can depend on the values of more important variables. The inference is parameterised by a set $\mathcal{Y}$ of subsets of V, but of most interest here is the simplest case when $\mathcal{Y}$ is the set of singleton subsets of V. It is easy to see that lexicographic models correspond with particular kinds of $\mathcal{Y}$-cp-trees in which the variable ordering, along with their associated value orderings, are identical in each branch of the cp-tree. Thus, for $\pi \in \mathcal{G}^{\mathrm{lex}}$ there exists a $\mathcal{Y}$-cp-tree σ with $\succcurlyeq_\sigma = \succcurlyeq_\pi$. This implies that, for any $\Gamma \subseteq \mathcal{L}$, and $\alpha, \beta \in \underline{V}$, if $\Gamma \models_{\mathcal{Y}} \alpha \geq \beta$ then $\Gamma \models^{\mathrm{lex}} \alpha \geq \beta$.

Proposition 2 suggests an approximation for lexicographic inference: define $\Gamma \models^{\mathrm{lex}}_* \alpha \geq \beta \iff \bar{\Gamma} \models_{\mathcal{Y}} \alpha \geq \beta$. A special case of this is used as the Lex-Basic inference method in [20]. If $\bar{\Gamma} \models_{\mathcal{Y}} \alpha \geq \beta$ then $\bar{\Gamma} \models^{\mathrm{lex}} \alpha \geq \beta$, and thus, using Proposition 2, if $\Gamma \models^{\mathrm{lex}}_* \alpha \geq \beta$ then $\Gamma \models^{\mathrm{lex}} \alpha \geq \beta$.

Inference based on additive utility models: A common assumption in multi-criteria reasoning, in particular in Multi-Attribute Utility Theory [10], is that an agent's utility function can be decomposed as a sum of unary functions. Write the set of variables V as $\{X_1, \ldots, X_n\}$. Define an *Additive Utility model* (over V), abbreviated to *AU-model*, to be a vector of (unary) functions $F = (f_1, \ldots, f_n)$, where, for $i = 1, \ldots, n$, f_i is a real-valued function on $\underline{X_i}$. Let $\mathcal{G}^{\mathrm{AU}}$ be the set of all AU models over V. For outcome α, define $F(\alpha)$ to be $\sum_{i=1}^n f_i(\alpha(X_i))$. We define $\succcurlyeq_F$ on $\underline{V} \times \underline{V}$ by $\alpha \succcurlyeq_F \beta \iff F(\alpha) \geq F(\beta)$. The corresponding relation $\models^{\mathrm{AU}}$ is given by $\Gamma \models^{\mathrm{AU}} \alpha \geq \beta$ if and only if $\alpha \succcurlyeq_F \beta$ for all $F \in \mathcal{G}^{\mathrm{AU}}$ such that $F \models \Gamma$, where the latter means $\alpha \succcurlyeq_F \beta$ for all $(\alpha, \beta) \in \Gamma^*$. This kind of inference is used e.g., in [7, 21, 18]. As is well known, since $\underline{V}$ is finite, a lexicographic ordering over V can be generated using an AU model, i.e., for $\pi \in \mathcal{G}^{\mathrm{lex}}$ there exists $F \in \mathcal{G}^{\mathrm{AU}}$ with $\succcurlyeq_F = \succcurlyeq_\pi$. Thus, if $\Gamma \models^{\mathrm{AU}} \alpha \geq \beta$ then $\Gamma \models^{\mathrm{lex}} \alpha \geq \beta$.

2.3 Example

Consider a system helping a user find a hotel in a particular city. Let us assume, for simplicity, only three attributes of interest, *Quality*, which takes values budget, medium and luxury; *Location*, which takes values city (representing that the hotel is in the city centre) and $\neg$city, and *Pool*, with values pool and $\neg$pool. The user indicates that she prefers the Morgan Hotel over the Three Bells hotel, where the former is a luxury hotel in the city centre but without a swimming pool, and the latter is a medium quality hotel outside the centre with a pool. The system induces a preference φ_1 of $(\mathsf{luxury}, \mathsf{city}, \neg\mathsf{pool})$ over $(\mathsf{medium}, \neg\mathsf{city}, \mathsf{pool})$. The system also has a general preference rule φ_2 representing the assumption that having a pool is at least as good as not having one *ceteris paribus* (all else being equal), so that φ_2 is the preference statement $\mathsf{pool} \geq \neg\mathsf{pool} \parallel \{\mathit{Quality}, \mathit{Location}\}$. From φ_2 we induce a preference of (q, l, pool) over $(q, l, \neg\mathsf{pool})$ for each value q of *Quality*, and each value l of *Location*. Assuming transitivity of preferences, we can infer from φ_1 and φ_2 that $(\mathsf{luxury}, \mathsf{city}, \neg\mathsf{pool}) \geq (\mathsf{medium}, \neg\mathsf{city}, \neg\mathsf{pool})$.

The user also indicates a preference φ_3 for $(\mathsf{medium}, \mathsf{city}, \mathsf{pool})$ over $(\mathsf{luxury}, \neg\mathsf{city}, \mathsf{pool})$. If we assume that the user's underlying preference model is a lexicographic order then we can infer from inputs $\{\varphi_1, \varphi_2, \varphi_3\}$, that (for example) $(\mathsf{medium}, \mathsf{city}, \neg\mathsf{pool})$ is preferred to $(\mathsf{luxury}, \neg\mathsf{city}, \mathsf{pool})$, and so we might show option $(\mathsf{medium}, \mathsf{city}, \neg\mathsf{pool})$ to the user before we show them $(\mathsf{luxury}, \neg\mathsf{city}, \mathsf{pool})$. The reason for the inference is that φ_1 and φ_3 imply that *Quality* is not the most important variable, and φ_1 and φ_2 imply that *Pool* cannot be the most important variable, so *Location* is the only possible most important variable. φ_1 then implies that city is preferred to $\neg$city. Thus, any lexicographic model of $\{\varphi_1, \varphi_2, \varphi_3\}$ has $(\mathsf{medium}, \mathsf{city}, \neg\mathsf{pool})$ preferred to $(\mathsf{luxury}, \neg\mathsf{city}, \mathsf{pool})$.

Note that this inference is not made with the Additive Utility approach. Consider, for example, the AU model $F = (f_1, f_2, f_3)$ given by: $f_1(\mathsf{luxury}) = 2$, $f_2(\mathsf{city}) = 3$ and $f_3(\mathsf{pool}) = 4$, with the other value(s) of each f_i being all zero. Then F satisfies φ_1, φ_2 and φ_3. But $F(\mathsf{medium}, \mathsf{city}, \neg\mathsf{pool}) = 3 < F(\mathsf{luxury}, \neg\mathsf{city}, \mathsf{pool}) = 6$. This implies that $\{\varphi_1, \varphi_2, \varphi_3\} \not\models^{\mathrm{AU}} (\mathsf{medium}, \mathsf{city}, \neg\mathsf{pool}) \geq (\mathsf{luxury}, \neg\mathsf{city}, \mathsf{pool})$.

An example of an inference that is made from $\{\varphi_1, \varphi_2, \varphi_3\}$ with $\models^{\mathrm{lex}}$ but not with $\models_{\mathcal{Y}}$ from [23] or with $\models^{\mathrm{lex}}_*$ (see Section 2.2 above) is $(\mathsf{budget}, \mathsf{city}, \mathsf{pool}) \geq (\mathsf{luxury}, \neg\mathsf{city}, \mathsf{pool})$, illustrating that $\models^{\mathrm{lex}}_*$ is not generally equal to $\models^{\mathrm{lex}}$.

Suppose we have an additional user preference φ_4, equalling $(\mathsf{luxury}, \mathsf{city}, \neg\mathsf{pool}) \geq (\mathsf{medium}, \mathsf{city}, \mathsf{pool})$. Any lexicographic model π satisfying $\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ will have *Quality* as the second most important variable after *Location*. It might be assumed that the values of *Quality* are single-peaked with respect to the ordering budget, medium, luxury, i.e., medium is not the worst value of *Quality*. If we add this extra assumption, then we can now deduce, for example, $(\mathsf{medium}, \neg\mathsf{city}, \mathsf{pool}) \geq (\mathsf{budget}, \neg\mathsf{city}, \mathsf{pool})$.

3 SINGLETON LEX INFERENCE

In this section we consider an even simpler kind of user preference model, where outcomes are compared just on their values on a single variable, using some total order on those values. This leads to an even more adventurous form of preference inference, but which has the same inferences $\alpha \geq \beta$ when α and β differ on every variable.

A singleton lex model (over set of variables V) is defined to be a pair $(Y, \succcurlyeq_Y)$, where $Y \in V$, and $\succcurlyeq_Y$ is a total order on $\underline{Y}$. Let $\mathcal{G}^{\mathrm{SL}}$ be the set of all singleton lex models (over V). For $\tau = (Y, \geq_Y)$, define the relation $\succcurlyeq_\tau$ on $\underline{V}$ by: $\alpha \succcurlyeq_\tau \beta \iff \alpha(Y) \geq_Y \beta(Y)$. Thus relation $\succcurlyeq_\tau$ compares two outcomes by considering only their values on variable Y.

We define, for $\mathfrak{R} \subseteq \underline{V} \times \underline{V}$, $\tau \models \mathfrak{R} \iff \succcurlyeq_\tau \supseteq \mathfrak{R}$, i.e., $\alpha \succcurlyeq_\tau \beta$ for all $(\alpha, \beta) \in \mathfrak{R}$. For set of preference statements, $\Gamma \subseteq \mathcal{L}$, we define $\tau \models \Gamma \iff \tau \models \Gamma^*$. We define the inference relation $\models^{\mathrm{SL}}$ from this in the usual way. $\Gamma \models^{\mathrm{SL}} \alpha \geq \beta$ if and only if $\alpha \succcurlyeq_\tau \beta$ holds for all $\tau \in \mathcal{G}^{\mathrm{SL}}$ such that $\tau \models \Gamma$. The induced relation on outcomes, $\succcurlyeq^{\mathrm{SL}}_\Gamma$, is given by $\alpha \succcurlyeq^{\mathrm{SL}}_\Gamma \beta \iff \Gamma \models^{\mathrm{SL}} \alpha \geq \beta$. Relation $\succcurlyeq^{\mathrm{SL}}_\Gamma$ is a pre-order containing Γ^*.

3.1 Computing singleton lex inference

The result below shows that there is a simple characterisation of this preference inference, leading to an efficient algorithm.

Proposition 3 *Consider any $\Gamma \subseteq \mathcal{L}$ and any $\alpha, \beta \in \underline{V}$. Then $\Gamma \models^{\mathrm{SL}} \alpha \geq \beta$ holds if and only if for all $Y \in V$ either $\Gamma^{\downarrow Y}$ has a cycle[2] or $\alpha(Y) = \beta(Y)$ or $(\alpha(Y), \beta(Y))$ is in the transitive closure of $\Gamma^{\downarrow Y}$.*

Proposition 3 shows that the relation $\succcurlyeq^{\mathrm{SL}}_\Gamma$ is similar to a Pareto ordering, since there is an independent condition that is tested for each variable: $\alpha \succcurlyeq^{\mathrm{SL}}_\Gamma \beta \iff$ for each $Y \in V$, $\alpha(Y) \geq^Y_\Gamma \beta(Y)$, where $\geq^Y_\Gamma$ is equal to the reflexive and transitive closure of $\Gamma^{\downarrow Y}$ if the latter is acyclic, and otherwise, $\geq^Y = \underline{Y} \times \underline{Y}$.

Complexity of singleton lex inference: Based on Proposition 3, determining if $\Gamma \models^{\mathrm{SL}} \alpha \geq \beta$, for a given Γ and (α, β), can be performed in $O(n|\Gamma|\bar{d})$, where $\bar{d} = \frac{1}{n}\sum_{Y \in V} |\underline{Y}|$ is the mean domain size.

Sound and complete proof theory

Another characterisation of $\models^{\mathrm{SL}}$ is given by the proof theory below. Although not necessarily useful computationally, it gives some intuition about the nature of the inference. It can be shown that $\Gamma \models^{\mathrm{SL}} \alpha \geq \beta$ if and only if $\alpha \geq \beta$ can be deduced using the Reflexivity axiom, and Transitivity, Crossing and Irrelevant Variable inference rules (as defined below), from Γ^* (i.e., with axioms $\alpha \geq \beta$ for any $(\alpha, \beta) \in \Gamma^*$). In the rules below, α, β, γ and δ are outcomes and X is a variable. For $x \in \underline{X}$, $\alpha[X := x]$ is defined to be the outcome α' with $\alpha'(X) = x$ and $\alpha'(Y) = \alpha(Y)$ for all $Y \neq X$.

Reflexivity: $\alpha \geq \alpha$ for all $\alpha \in \underline{V}$.
Transitivity: From $\alpha \geq \beta$ and $\beta \geq \gamma$ deduce $\alpha \geq \gamma$.
Crossing: From $\alpha \geq \beta$ and $\gamma \geq \delta$
deduce $\alpha[X := \gamma(X)] \geq \beta[X := \delta(X)]$.

[2] Relation $\rhd$ on A is said to *have a cycle* if there exists $k > 2$, and $a_1, \ldots, a_k \in A$ with $a_k = a_1$ and for all $i = 1, \ldots, k-1$, $a_i \rhd a_{i+1}$ and $a_i \neq a_{i+1}$. If $\rhd$ does not have a cycle it is said to be *acyclic*.

Irrelevant Variable: From $\alpha \geq \beta$, $\gamma \geq \delta$ and $\delta \geq \gamma$ such that $\gamma(X) \neq \delta(X)$ deduce $\alpha[X := x] \geq \beta[X := x']$ for any $x, x' \in \underline{X}$.

The output of the Crossing inference rule is a kind of cross of the inputs. For variable X satisfying the conditions of the Irrelevant Variable inference, the values of X are irrelevant, i.e., changing them produces an equivalent pair of outcomes.

Theorem 1 $\Gamma \models^{\mathrm{SL}} \alpha \geq \beta$ *if and only if* $\alpha \geq \beta$ *can be deduced from* Γ^* *using the Reflexivity axiom, and the Transitivity, Crossing and Irrelevant Variable inference rules.*

3.2 Relationship between lex and SL inference

It is clear that a singleton lex model is a special case of a lexicographic model, which immediately implies that the $\models^{\mathrm{SL}}$ inference is at least as strong as the $\models^{\mathrm{lex}}$ inference: if $\Gamma \models^{\mathrm{lex}} \alpha \geq \beta$ then $\Gamma \models^{\mathrm{SL}} \alpha \geq \beta$. Also, if outcomes α and β differ on every variable then only the first $(Y, \geq_Y)$ pair in any lexicographic model is relevant; this implies that, for such α and β, we have $\Gamma \models^{\mathrm{lex}} \alpha \geq \beta \iff \Gamma \models^{\mathrm{SL}} \alpha \geq \beta$. This means that Proposition 3 can be used to determine if $\Gamma \models^{\mathrm{lex}} \alpha \geq \beta$ for such pairs (α, β). For more general pairs, the problem is harder, and a more sophisticated approach is needed, as shown in Section 4.

To see that $\models^{\mathrm{SL}}$ can enable strictly more inferences than $\models^{\mathrm{lex}}$: we have e.g., that $\{\varphi_1\} \models^{\mathrm{SL}} \alpha \geq \beta$ but $\{\varphi_1\} \not\models^{\mathrm{lex}} \alpha \geq \beta$, where φ_1 is $(\mathsf{luxury}, \mathsf{city}, \neg\mathsf{pool}) \geq (\mathsf{medium}, \neg\mathsf{city}, \mathsf{pool})$, and $\alpha = (\mathsf{luxury}, \mathsf{city}, \neg\mathsf{pool})$ and $\beta = (\mathsf{medium}, \mathsf{city}, \mathsf{pool})$.

Consistency w.r.t. $\mathcal{G}^{\mathrm{lex}}$ and $\mathcal{G}^{\mathrm{SL}}$: For $\Gamma \subseteq \mathcal{Y}$ we say that set of preference statements Γ *is consistent with respect to* $\mathcal{G}^{\mathrm{lex}}$ if there exists some model $\pi \in \mathcal{G}^{\mathrm{lex}}$ satisfying Γ, i.e., such that $\pi \models \Gamma$. Similarly, Γ *is consistent with respect to* $\mathcal{G}^{\mathrm{SL}}$ if there exists $\tau \in \mathcal{G}^{\mathrm{SL}}$ such that $\tau \models \Gamma$. It follows that Γ is not consistent with respect to $\mathcal{G}^{\mathrm{SL}}$ if and only if $\models^{\mathrm{SL}} = \underline{V} \times \underline{V}$, which, using Proposition 3, is if and only if $\Gamma^{\downarrow Y}$ has a cycle for all $Y \in V$.

Now, suppose that $\pi \models \Gamma$ for some lexicographic model π, which we write as $(Y_1, \geq_{Y_1}), \ldots, (Y_k, \geq_{Y_k})$. It is easy to see that $\tau \models \Gamma$, where $\tau = (Y_1, \geq_{Y_1})$. Conversely, if $\tau = (Y_1, \geq_{Y_1}) \in \mathcal{G}^{\mathrm{SL}}$ is such that $\tau \models \Gamma$ then the corresponding element $(Y_1, \geq_{Y_1})$ in $\mathcal{G}^{\mathrm{lex}}$ also satisfies Γ. This shows that Γ is consistent with respect to $\mathcal{G}^{\mathrm{lex}}$ if and only if Γ is consistent with respect to $\mathcal{G}^{\mathrm{SL}}$, if and only if there exists some $Y \in V$ with $\Gamma^{\downarrow Y}$ acyclic.

A slightly more general model: One can get a slight variation on $\models^{\mathrm{SL}}$ by allowing models $(Y, \geq)$ with $\geq$ being total pre-orders (i.e., transitive and complete relations, also known as weak orders). Call this relation $\models^{\mathrm{SLW}}$. The characterisation is then slightly simpler than Proposition 3: $\Gamma \models^{\mathrm{SLW}} \alpha \geq \beta$ holds if and only if for all $Y \in V$ either $\alpha(Y) = \beta(Y)$ or $(\alpha(Y), \beta(Y))$ is in the transitive closure of $\Gamma^{\downarrow Y}$. The Reflexivity axiom, and the Transitivity and Crossing inference rules form a sound and complete proof theory for $\models^{\mathrm{SLW}}$.

4 COMPUTING LEXICOGRAPHIC PREFERENCE INFERENCE

In this section we show how to compute, in polynomial time, lexicographic preference inference, i.e., determining, whether a given pair $\alpha \geq \beta$ is entailed by a set of comparative preference statements Γ. By definition, $\Gamma \models^{\mathrm{lex}} \alpha \geq \beta$ holds if and only if $\alpha \succcurlyeq_\pi \beta$ holds for every lex model π satisfying Γ. The set Γ therefore acts as a restriction on possible lexicographic models. Lemma 1 below shows that to generate a lexicographic model $(Y_1, \geq_{Y_1}), \ldots, (Y_k, \geq_{Y_k})$ satisfying Γ we need, for each i to satisfy a constraint on the choice of pair Y_i and $\geq_{Y_i}$ which depends on the earlier variables A_i, i.e., that $\geq_{Y_i} \supseteq \Gamma^{\downarrow Y_i}_{A_i}$. This leads naturally to an abstraction based on what we call a Next Variable Predicate (NVP), which restricts the choice of pair $(Y, \geq_Y)$ given previously chosen variables A, and hence restricts the set of lexicographic models π. An NVP $\mathcal{P}$ then entails $\alpha \geq \beta$ if $\alpha \succcurlyeq_\pi \beta$ holds for every π satisfying $\mathcal{P}$. We show that there is a simple algorithm for determining this, when the NVP satisfies a monotonicity property. We show that π satisfying Γ can be expressed as a monotonic NVP, thus allowing efficient testing of $\Gamma \models^{\mathrm{lex}} \alpha \geq \beta$. Other natural inputs can also be expressed as monotonic NVPs, in particular, restrictions on variable and value orderings. This enables efficient inference if we have a mixture of inputs, such as a set of comparative preference statements, along with restrictions on variable and local value orderings.

Lemma 1 *Let* π *equalling* $(Y_1, \geq_1), \ldots, (Y_k, \geq_k)$ *be a lexicographic model. For* $i = 1, \ldots, k$, *define* $A_i = \{Y_1, \ldots, Y_{i-1}\}$. *Let* α *and* β *be two outcomes, and let* $\Gamma \subseteq \mathcal{L}$ *be a set of preference statements. Then*

(i) $\alpha \succcurlyeq_\pi \beta$ *if and only if for all* $i = 1, \ldots, k$, *[*$\alpha(A_i) = \beta(A_i) \Rightarrow \alpha(Y_i) \geq_i \beta(Y_i)$*].*

(ii) $\pi \models \Gamma$ *if and only if for all* $i = 1, \ldots, k$, $\geq_i \supseteq \Gamma^{\downarrow Y_i}_{A_i}$.

4.1 Next Variable Predicates (NVPs)

We formalise the notion of NVPs, and we show that a wide range of natural restrictions can be expressed in terms of monotonic NVPs.

A Next-Variable Predicate (abbreviated to NVP) $\mathcal{P}$ is a relation on triples of the form $(A, Y, \geq_Y)$, where $A \subseteq V$, $Y \in V - A$ and $\geq_Y$ is a total order on $\underline{Y}$. Let π be a lexicographic model $(Y_1, \geq_{Y_1}), \ldots, (Y_k, \geq_{Y_k})$. Then, π is said to satisfy NVP $\mathcal{P}$ (also written as $\pi \models \mathcal{P}$) if for all $i = 1, \ldots, k$, $\mathcal{P}(A_i, Y_i, \geq_{Y_i})$ holds, where $A_i = \{Y_1, \ldots, Y_{i-1}\}$. For NVP $\mathcal{P}$, we define $\mathcal{P} \models^{\mathrm{lex}} \alpha \geq \beta$ if $\pi \models \alpha \geq \beta$ for all lexicographic models π satisfying $\mathcal{P}$.

Monotonic NVP: NVP $\mathcal{P}$ is said to be monotonic if for all A, B such that $A \subseteq B \subseteq V$, for all $Y \in V - B$ and for all total orders $\geq_Y$ on $\underline{Y}$, $\mathcal{P}(A, Y, \geq_Y) \Rightarrow \mathcal{P}(B, Y, \geq_Y)$.

Boolean operations on NVPs can be defined in the obvious way. In particular, for NVPs $\mathcal{P}_1$ and $\mathcal{P}_2$, triple $(A, Y, \geq_Y)$ is defined to satisfy $\mathcal{P}_1 \wedge \mathcal{P}_2$ if and only if $(A, Y, \geq_Y)$ satisfies both $\mathcal{P}_1$ and $\mathcal{P}_2$. Also, triple $(A, Y, \geq_Y)$ is defined to satisfy $\mathcal{P}_1 \vee \mathcal{P}_2$ if and only if $(A, Y, \geq_Y)$ satisfies either $\mathcal{P}_1$ or $\mathcal{P}_2$.

Proposition 4 *If* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *are monotonic NVPs then* $\mathcal{P}_1 \wedge \mathcal{P}_2$ *and* $\mathcal{P}_1 \vee \mathcal{P}_2$ *are both monotonic NVPs.*

This key result (which follows immediately from the definitions) means that any NVP that is built, using conjunctions and disjunctions from monotonic NVPs, is also monotonic.

4.2 NVPs for preference statements, variable and value orderings

We show how to generate a NVP corresponding to a set Γ of comparative preference statements, and also NVPs corresponding to restrictions on variable and value orderings.

Expressing basic variable ordering restrictions: Let X and X' be different variables in V, and let π be the lexicographic model $(Y_1, \geq_{Y_1}), \ldots, (Y_k, \geq_{Y_k})$. We say that π satisfies $X \rhd X'$ if the following holds: if $X' = Y_i$ for some $i = 1, \ldots, k$ then there exists $j < i$ with $Y_j = X$. Thus π satisfies $X \rhd X'$ if X appears earlier in π than X' when X' appears in π.

Expressing basic value ordering restrictions: Let x and x' be two values of variable X ($\in V$). We say that π satisfies $x \geq x'$ if the following holds: for all $i = 1, \ldots, k$, if $Y_i = X$ then $x \geq_{Y_i} x'$.

Definition 1 *Consider an arbitrary triple $(A, Y, \geq_Y)$, where $A \subseteq V$, $Y \in V - A$ and $\geq_Y$ is a total order on $\underline{Y}$. For $\Gamma \subseteq \mathcal{L}$, different variables $X, X' \in V$ and $x, x' \in \underline{X}$, we define the NVPs $\mathcal{P}_{\models\Gamma}$, $\mathcal{P}_{X \rhd X'}$ and $\mathcal{P}_{x \geq x'}$ as follows.*

- *$\mathcal{P}_{\models\Gamma}(A, Y, \geq_Y)$ if and only if $\geq_Y \supseteq \Gamma_A^{\downarrow Y}$.*
- *$\mathcal{P}_{X \rhd X'}(A, Y, \geq_Y)$ if and only if either $A \ni X$ or $Y \neq X'$.*
- *$\mathcal{P}_{x \geq x'}(A, Y, \geq_Y)$ if and only if either $Y \neq X$ or $x \geq_Y x'$.*

The result below shows that the NVPs defined in Definition 1 express their intended meaning.

Proposition 5 *Let π be an arbitrary lexicographic model. For $\Gamma \subseteq \mathcal{L}$, different variables $X, X' \in V$ and $x, x' \in \underline{X}$, NVPs $\mathcal{P}_{\models\Gamma}$, $\mathcal{P}_{X \rhd X'}$ and $\mathcal{P}_{x \geq x'}$ are all monotonic, and*

- *$\pi \models \mathcal{P}_{\models\Gamma} \iff \pi \models \Gamma$;*
- *$\pi \models \mathcal{P}_{X \rhd X'} \iff \pi \models X \rhd X'$; and*
- *$\pi \models \mathcal{P}_{x \geq x'} \iff \pi \models x \geq x'$.*

Arbitrary restrictions on value orderings can be represented using disjunctions and conjunctions of NVPs of the form $\mathcal{P}_{x \geq x'}$, and similarly for restrictions of variable orderings based on $\mathcal{P}_{X \rhd X'}$. Thus if we have as inputs a set of preference statements $\Gamma \subseteq \mathcal{L}$, and restrictions on value and variable orderings, these inputs can be represented by a monotonic NVP, because of Propositions 5 and 4.

4.3 Algorithm for determining lexicographic entailment from Monotonic NVP

A natural idea for an algorithm for constructing a lexicographic model satisfying Γ but not $\alpha \geq \beta$ (thus showing that $\Gamma \not\models^{\text{lex}} \alpha \geq \beta$) would involve backtracking search over different variable orderings. However, such an algorithm would presumably be exponential in the worst case. In fact, the monotonicity property of the NVP allows a polynomial backtrack-free algorithm. The algorithm is similar to those for preference entailment based on conditional lexicographic orders described in [22] and [23], and the idea behind the proof is also similar, although with some different technical issues.

The idea behind the algorithm is as follows. We're trying to construct a lexicographic model π that satisfies $\mathcal{P}$ but not $\alpha \geq \beta$, so that $\beta \succ_\pi \alpha$ (i.e., $\alpha \not\succcurlyeq_\pi \beta$). We build up π incrementally, choosing $(Y_1, \geq_{Y_1})$ first and then $(Y_2, \geq_{Y_2})$, and so on. Suppose that we have picked already $Y_1, \ldots, Y_{j-1}$, where $\alpha(Y_i) = \beta(Y_i)$ for each $i \in \{1, \ldots, j-1\}$, and let $A_j = \{Y_1, \ldots, Y_{j-1}\}$. At each stage we see if there is another variable Y and ordering $\geq_Y$ that we can choose with $\alpha(Y) \not\geq_Y \beta(Y)$ and such that $\mathcal{P}(A_j, Y, \geq_Y)$ holds. If so, then we have constructed a lexicographic model that satisfies $\mathcal{P}$ but not $\alpha \geq \beta$, proving that $\mathcal{P} \not\models \alpha \geq \beta$. If this is not possible, we choose, if possible, any Y and $\geq_Y$ such that $\mathcal{P}(A_j, Y, \geq_Y)$ holds and $\alpha(Y) = \beta(Y)$, and let $Y_j = Y$ and $\geq_j$ equal $\geq_Y$.

procedure Does $\mathcal{P}$ lexicographically entail $\alpha \geq \beta$?
if $\alpha = \beta$ **then** return **true** and **stop**;
for $j := 1, \ldots, n$
  let A_j equal $A_j = \{Y_1, \ldots, Y_{j-1}\}$ (in particular, $A_1 = \emptyset$);
  if there exists $Y \in V - A_j$ and $\geqslant$ such that $\mathcal{P}(A_j, Y, \geqslant)$ holds
    and $\alpha(Y) \not\geqslant \beta(Y)$
    then return **false** and **stop**;
  if there exists $Y \in V - A_j$ and $\geqslant$ such that $\mathcal{P}(A_j, Y, \geqslant)$ holds
    and $\alpha(Y) = \beta(Y)$
    then let $Y_j = Y$ and $\geqslant_j = \geqslant$ (for any such pair Y and $\geqslant$)
    else return **true** and **stop**;
  next j;
return **true**.

4.3.1 Correctness of algorithm

The theorem states the correctness of the algorithm.

Theorem 2 *Let $\mathcal{P}$ be a monotonic Next-Variable Predicate, and let $\alpha, \beta \in \underline{V}$ be outcomes. The above procedure for lexicographic inference is correct, i.e., it returns **true** if $\mathcal{P} \models^{\text{lex}} \alpha \geq \beta$ and it returns **false** otherwise.*

Decisive Sequence: A Lexicographic model $(Y_1, \geq_1), \ldots, (Y_k, \geq_k)$ is said to be a decisive sequence with respect to $\mathcal{P}$ and $\alpha \geq \beta$ if $\alpha(Y_k) \not\geq_k \beta(Y_k)$, and for $j = 1, \ldots, k-1$, $\alpha(Y_j) = \beta(Y_j)$, and, for $j = 1, \ldots, k$, $\mathcal{P}(A_j, Y_j, \geq_j)$ holds, where $A_j = \{Y_1, \ldots, Y_{j-1}\}$. Thus, if π is a decisive sequence with respect to $\mathcal{P}$ and $\alpha \geq \beta$, then π satisfies $\mathcal{P}$ but not $\alpha \geq \beta$.

The next lemma follows easily from the definition of lexicographic inference, and the following lemma sums up some easy observations about the algorithm.

Lemma 2 *$\mathcal{P} \not\models^{\text{lex}} \alpha \geq \beta$ if and only if there exists a decisive sequence with respect to $\mathcal{P}$ and $\alpha \geq \beta$.*

Lemma 3 *Let $Y_1, \ldots, Y_k$ be the sequence of sets generated by the algorithm, and let $\geq_1, \ldots, \geq_k$ be the sequence of orderings.*

*(i) The algorithm returns **true** if and only if it does not return **false**.*
(ii) For $j = 1 \ldots, k-1$, $\alpha(Y_j) = \beta(Y_j)$.
*(iii) If the algorithm returns **false** then $(Y_1, \geq_1), \ldots, (Y_k, \geq_k)$ is a decisive sequence with respect to $\mathcal{P}$ and $\alpha \geq \beta$.*

4.3.2 Proof of Theorem 2

By Lemma 3(i), the algorithm returns **true** if and only if it does not return **false**. Thus, to prove the result, it is sufficient to show that the algorithm returns **false** if and only if $\mathcal{P} \not\models^{\text{lex}} \alpha \geq \beta$.

First let us assume that the algorithm returns **false**. Then the sequence $(Y_1, \geq_1), \ldots, (Y_k, \geq_k)$ generated by the algorithm is a decisive sequence, by Lemma 3(iii), and thus $\mathcal{P} \not\models^{\text{lex}} \alpha \geq \beta$, by Lemma 2.

Conversely, let us assume that $\mathcal{P} \not\models^{\text{lex}} \alpha \geq \beta$, and so there exists a decisive sequence, $(X_1, \geq'_1), \ldots, (X_l, \geq'_l)$, by Lemma 2. Thus $\alpha(X_l) \neq \beta(X_l)$. To prove a contradiction, assume that the algorithm does not return **false** (and thus returns **true** by Lemma 3(i)). Let $Y_1, \ldots, Y_k$ be the sequence of sets generated by the algorithm.

First consider the case where every X_i is in $\{Y_1, \ldots, Y_k\}$, and consider j such that $Y_j = X_l$. Thus $\alpha(Y_j) \neq \beta(Y_j)$ which implies that $j = k$, by Lemma 3(ii). Let $A'_l = \{X_1, \ldots, X_{l-1}\}$. By definition of a decisive sequence, $\mathcal{P}(A'_l, X_l, \geq'_l)$ holds and $\alpha(X_l) \not\geq'_l$

$\beta(X_l)$. Now, $Y_k = X_l \notin A'_l$, so $A'_l \subseteq \{Y_1, \ldots, Y_{k-1}\} = A_k$. Monotonicity of NVP $\mathcal{P}$ implies that $\mathcal{P}(A_k, X_l, \geq'_l)$ holds, which means that the algorithm would return **false**, which contradicts the assumption.

Now, let us consider the other case, where there exists some X_i which is not in $\{Y_1, \ldots, Y_k\}$, and consider a minimal such i. Let $A'_i = \{X_1, \ldots, X_{i-1}\}$. By definition of a decisive sequence, $\mathcal{P}(A'_i, X_i, \geq'_i)$ holds. We have $A'_i \subseteq \{Y_1, \ldots, Y_k\}$, and so monotonicity of $\mathcal{P}$ implies that $\mathcal{P}(A_{k+1}, X_i, \geq'_i)$ holds, where we define $A_{k+1} = \{Y_1, \ldots, Y_k\}$. Also, $\alpha(X_i) = \beta(X_i)$ (if $i < l$) or $\alpha(X_i) \not\geq'_i \beta(X_i)$ (if $i = l$), by definition of a decisive sequence. Now, X_i and $\geq'_i$ satisfy the conditions that enable the algorithm to choose another variable Y $(= X_i)$, which contradicts Y_k being the last variable generated by the algorithm. □

4.4 Application for $\Gamma \models^{\text{lex}} \alpha \geq \beta$

To test if $\Gamma \models^{\text{lex}} \alpha \geq \beta$ we use $\mathcal{P}_{\models\Gamma}$ as $\mathcal{P}$ in the algorithm. The algorithm can then be somewhat simplified: the conditions of the second and third **if** statements can be replaced, respectively, by the following two statements:

if $\exists\, Y_j \in V - A_j$ such that $\Gamma^{\downarrow Y_j}_{A_j} \cup \Big\{ \Big(\beta(Y_j), \alpha(Y_j)\Big) \Big\}$ is acyclic

if $\exists\, Y \in V - A_j$ such that $\alpha(Y) = \beta(Y)$ and $\Gamma^{\downarrow Y_j}_{A_j}$ is acyclic

Complexity of determining if $\Gamma \models^{\text{lex}} \alpha \geq \beta$: A careful implementation of the algorithm allows a complexity of $O(n^2|\Gamma|\bar{d})$, with $\bar{d}$ being the mean domain size.

5 CONCLUSIONS AND DISCUSSION

We have shown how lexicographic preference inference (as well as singleton lex inference) can be computed with a low-order polynomial algorithm. Propositions 4 and 5 and Theorem 2 mean that the algorithm can be applied to compute lexicographic inference based on a wide (and mixed) range of inputs; this can include restrictions on the value and variable orderings, as well as an input set of preferences statements with a relatively general language. For example, the inputs could include assumptions that certain domains are single-peaked [8], and it could also include restrictions on the variable ordering such as that the most important variable is either X_1, X_3 or X_6.

Lexicographic preference inference may be appropriate when the inputs are relatively weak. There are also variations of this family of preference relations that might well be considered, such as lexicographic models with the local orderings $\geq_Y$ being total pre-orders rather than total orders, or where Y can be a small set of variables (analogously to the conditional lexicographic models in [23]). It would be interesting to see if the approach in this paper could be adapted, and also if the inference technique in [23] could be adapted to still more general kinds of input, including restrictions on variable and value orderings.

ACKNOWLEDGEMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

REFERENCES

[1] M. Bienvenu, J. Lang, and N. Wilson, 'From preference logics to preference languages, and back', in *Proc. KR 2010*, (2010).

[2] R. Booth, Y. Chevaleyre, J. Lang, J. Mengin, and C. Sombattheera, 'Learning conditionally lexicographic preference relations', in *ECAI*, pp. 269–274, (2010).

[3] C. Boutilier, R. I. Brafman, C. Domshlak, H. Hoos, and D. Poole, 'CP-nets: A tool for reasoning with conditional *ceteris paribus* preference statements', *Journal of Artificial Intelligence Research*, **21**, 135–191, (2004).

[4] S. Bouveret, U. Endriss, and J.Lang, 'Conditional importance networks: A graphical language for representing ordinal, monotonic preferences over sets of goods', in *Proc. IJCAI-09*, pp. 67–72, (2009).

[5] R. Brafman, C. Domshlak, and E. Shimony, 'On graphical modeling of preference and importance', *Journal of Artificial Intelligence Research*, **25**, 389–424, (2006).

[6] M. Bräuning and E. Hüllermeier, 'Learning conditional lexicographic preference trees', in *Preference Learning (PL-12), ECAI-12 workshop*, (2012).

[7] D. Bridge and F. Ricci, 'Supporting product selection with query editing recommendations', in *RecSys '07*, pp. 65–72, New York, NY, USA, (2007). ACM.

[8] V. Conitzer, 'Eliciting single-peaked preferences using comparison queries', *J. Artif. Intell. Res. (JAIR)*, **35**, 161–191, (2009).

[9] J. Dombi, C. Imreh, and N. Vincze, 'Learning lexicographic orders', *European Journal of Operational Research*, **183**(2), 748756, (2007).

[10] J. Figueira, S. Greco, and M. Ehrgott, *Multiple Criteria Decision Analysis—State of the Art Surveys*, Springer International Series in Operations Research and Management Science Volume 76, 2005.

[11] P. Fishburn, 'Lexicographic orders, utilities and decision rules: A survey', *Management Science*, **20**(11), 1442–1471, (1974).

[12] P. Fishburn, 'Axioms for lexicographic preferences', *The Review of Economic Studies*, **42**(3), 415–419, (1975).

[13] P. A. Flach and E.T. Matsubara, 'A simple lexicographic ranker and probability estimator', in *ECML*, pp. 575–582, (2007).

[14] E. Freuder, R. Heffernan, R. Wallace, and N. Wilson, 'Lexicographically-ordered constraint satisfaction problems', *Constraints*, **15**(1), 1–28, (2010).

[15] J. Fürnkranz and E. Hüllermeier (eds.), *Preference Learning*, Springer-Verlag, 2010.

[16] J. Goldsmith, J. Lang, M. Truszczyński, and N. Wilson, 'The computational complexity of dominance and consistency in CP-nets', *Journal of Artificial Intelligence Research*, **33**, 403–432, (2008).

[17] J. Lang, 'Logical preference representation and combinatorial vote', *Ann. Mathematics and Artificial Intelligence*, **42**(1), 37–71, (2004).

[18] R. Marinescu, A. Razak, and N. Wilson, 'Multi-objective constraint optimization with tradeoffs', in *CP*, pp. 497–512, (2013).

[19] M. McGeachie and J. Doyle, 'Utility functions for ceteris paribus preferences', *Computational Intelligence*, **20**(2), 158–217, (2004).

[20] W. Trabelsi, N. Wilson, and D. Bridge, 'Comparative preferences induction methods for conversational recommenders', in *ADT*, pp. 363–374, (2013).

[21] W. Trabelsi, N. Wilson, D. Bridge, and F. Ricci, 'Preference dominance reasoning for conversational recommender systems: a comparison between a comparative preferences and a sum of weights approach', *International Journal on Artificial Intelligence Tools*, **20**(4), 591–616, (2011).

[22] N. Wilson, 'An efficient upper approximation for conditional preference', in *Proc. ECAI-06*, pp. 472–476, (2006).

[23] N. Wilson, 'Efficient inference for expressive comparative preference languages', in *Proc. IJCAI-09*, pp. 961–966, (2009).

[24] N. Wilson, 'Computational techniques for a simple theory of conditional preferences', *Artificial Intelligence*, **175**(7-8), 1053–1091, (2011).

[25] N. Wilson, 'Importance-based semantics of polynomial comparative preference inference', in *ECAI*, (2012).

[26] N. Wilson, *Preference Inference Based on Lexicographic Models (extended version of current paper including proofs)*, available at http://4c.ucc.ie/∼nwilson/public/LexInferenceProofs.pdf (clickable link: http://4c.ucc.ie/%7Enwilson/public/LexInferenceProofs.pdf), 2014.